AppSense Privilege management - Technical reviewers guide
Preface

This paper provides an in-depth discussion on AppSense Application Manager and how it works to help customers achieve a least privilege security stance in their organization.

For an overview and background on what this product does and the business value behind the concept of effective Privilege management, please refer to the whitepaper titled “AppSense Privilege management”.

Contents

Preface 3
The case for comprehensive Privilege management 3
How AppSense delivers on Privilege management 3
Common scenarios and capabilities 4
Privilege management and the Internet 4
Trusted Ownership 5
Understanding your existing local administrator landscape 6
Self-elevation 6
Granular control of self-elevation 7
Auditing & reporting of self-elevation 7
AppSense Application Manager and internal App stores 8
Secure dialog boxes 8
Child process control 8
Conclusion 9
Appendix 9
How it works: AppSense Application Manager and Local Security Authority (LSA) Tokens 9
Functional Matrix 10
Differences between AppSense Application Manager and AppSense Application Manager – User Rights Management Edition 11
The case for comprehensive Privilege management

Providing users with administrative privilege on their desktop provides them with access to areas of the desktop that, if misused, can result in high support costs and a compromised user experience. In many cases, security is compromised through the loss of data or attack from malicious software.

However, there are many scenarios where users do require local admin rights in order to be able to work effectively. Many applications, including newly released applications – allow changes to be made to hardware settings or network adapters and all require administrative privilege in order to execute. This also includes web application updates, the installation of Active-X components, Adobe/Flash/Java updates or installation of printer drivers.

For any of these common tasks, a user requires local administrative privilege on their desktop. Windows 7 attempted to solve this issue with its implementation of User Account Control (UAC) though does not solve the complete granular needs of enterprise management.

With AppSense Privilege management, the privilege level of a user, group or role can be elevated or reduced for application and control panel applets. A user with standard user access only to their PC can be given specific elevated rights to add a printer driver for their home printer (a task that requires administrator access to the machine). Alternatively, a user with administrative access to their Windows desktop can have their rights reduced for antivirus settings.

By controlling user privileges throughout the user session, IT can now provide users with the accessibility they require to perform their job, while protecting the desktop and the environment and reducing desktop management costs.

How AppSense delivers on Privilege management

The perfect balance between user productivity, security and lower desktop TCO is to control user privilege not at a session or account level, but instead, at an application or individual task level.

With AppSense Application Manager, access to applications and tasks is managed dynamically by managing user rights on-demand in response to user actions. The alternative is to revert back to the provision of a local admin account which ultimately increases the overall cost of support as the laptop falls vulnerable to misuse, malware and user downtime.

Elevated privilege can be applied to a named application or control panel applet for a particular user or user group, or reducing the rights of an administrative user to that of a standard user account. By controlling user privileges throughout the user session, IT can now provide users with the accessibility they require to perform their job, while protecting the desktop and the environment and reducing desktop management costs.

AppSense offers two solutions in this area to address Privilege management.

- The full functioned; AppSense Application Manager 8 delivers a world-class industry solution to:
  - Granularly control user rights and privileges
  - Manage application access
  - Implement whitelisting and blacklisting
  - Enforce device based licensing
  - Block untrusted software and code
  - Control network access

- A reduced functionality solution, AppSense Application Manager - User Rights Management edition that implements the ability to purely control user rights and privileges.

For more information and a detailed matrix on specific functionality, please refer to the appendix.
Common scenarios and capabilities

Elevating privileges for applications
**Scenario:** Specify applications that need to be run with elevated rights within a standard non-administrative user desktop session. This capability assists in remediating application compatibility scenarios relating to least privilege.

**Benefit:** Here, the user account or desktop session does not have admin rights; however, the user is able to run specific applications under an elevated context (with no explicit consent prompts required) when the application needs administrative rights to run. This is especially relevant if it is a legacy application written assuming that all users had admin rights (and a common scenario for Windows XP deployments).

Elevating privileges for control panel applets
**Scenario:** Many roaming users require the ability to install printers and change network and firewall settings; all require Control Panel applets to be accessed with admin rights.

**Benefit:** Privilege management can elevate privileges for individual applets, meaning a standard user can still make the required changes to their desktop in order to do their job without having administrative access to the whole desktop.

Reducing user rights privileges for applications and control panel applets
**Scenario:** Specify applications or control panel applets to run with reduced privileges. The user has admin rights by default, but specified applications such as Task Manager, Firewall settings or RegEdit are forced to run with reduced privileges as a non-administrative user.

**Benefit:** Useful where a business is forced to implement admin rights by default, such as in a legacy Windows XP deployment, though wants to ensure that certain system and applications cannot be changed.

Privilege management and the Internet

AppSense also extends the Privilege management capabilities through the control of web based software installations. IT teams can now enable non-administrative users to install and update pre-approved web-delivered software in a controlled manner without the need to grant full administrative user rights to the desktop. This increases user productivity while minimizing IT support cost.

**Website-level web installation rights**
Allowing end-users to initiate web-based software installations is a potentially risky proposition, even when it can be accomplished without granting full administrative rights. For every legitimate business application source there are countless other websites hosting malware or other software that are not approved for corporate use such as non-productivity or unlicensed software.

With AppSense Application Manager, IT administrators can “whitelist” approved websites from which users are automatically authorized to install approved software.

As an example, IT team can preauthorize web based software installations only from known sites such as [www.adobe.com](http://www.adobe.com) and [www.gotomeeting.com](http://www.gotomeeting.com). This provides end-users with immediate access to common business applications such as Adobe Reader, Adobe Air, Adobe Flash Player and the GoToMeeting or WebEx web conferencing clients without IT application delivery bottlenecks and inefficiencies.
In addition to enabling the user population to access required personal productive applications which are not delivered by IT, at the same time, security and integrity is maintained as users are blocked from performing web-based software installation from all other sources, reducing exposure to security vulnerabilities or system instability and associated costs through unauthorized software installation.

**Application level web installations**

While website-level software installation policies may represent a good balance between security and simplicity for many organizations, some organizations may require more granular control over the applications that users can install from the specifically approved websites.

To revisit the earlier example, an IT administrator may wish to allow installation of Adobe Reader but block Adobe Air, Adobe Flash Player and/or any other applications from www.adobe.com. In this scenario, AppSense Application Manager provides IT administrators with the required controls to accomplish this, providing the option to whitelist specific applications by version and ActiveX control class ID within the named website as needed. This provides the assurance that only trusted versions of specific applications may be installed from the web by end-users.

This also helps with managing application version control within enterprise environments.

**Pre-configured web installation rights application templates**

While AppSense Application Manager offers granular user rights policy creation and customization capabilities to address a wide range of complex scenarios, it also makes managing common application policy scenarios quite simple thereby reducing implementation time.

IT teams extending web-based software installation rights to their end-users can draw from a library of pre-defined templates for common business applications. Applying these templates saves time and provides the assurance that users are only gaining the minimum level of rights escalation required to be productive.

AppSense Application Manager includes the option to give select users the ability to “self-elevate” the execution of an application to include elevated privileges, in addition ensures that any application elevated remains secure, access to Windows Explorer or any connected dialogs remain as a standard user and do not inherit administrator rights from within the elevated application.

Self-elevation can be configured by the System administrator so that only a predefined list of authorized applications can be elevated, or alternatively and list of prohibited applications can be specified to prevent them from being self-elevated, e.g. CMD.exe or RegEdit.exe

**Trusted Ownership**

Trusted Ownership prevents the execution of any code, even unknown, introduced by a non-trusted owner (example: a typical user account). This unique concept of “Trusted Ownership” is a key element of the AppSense application entitlement policy management approach. Trusted Ownership provides an additional degree of security and control by only allowing those applications that are (associated) “owned” by a pre-defined and trusted user account to execute. An example of a “trusted owner” is the Local System, administrator or Domain administrator account and typically an account that manages the desktop.

The AppSense Application Manager capabilities are fully compatible with the existing AppSense Trusted Ownership policy framework. IT administrators may specify that approved web-based installations such as Adobe Reader and GoToMeeting will be automatically configured so that file ownership is assigned to an existing trusted owner account rather than the initiating user’s account. This allows approved web-installed applications to execute even in environments with tightly managed Trusted Ownership policies.
Understanding your existing local administrator landscape

Removing unnecessary local admin rights is a major objective for many IT departments as it lowers the overall cost of maintaining a desktop environment while also intelligently ensuring users have access to the applications and OS features they require to remain as productive as possible.

Unfortunately, without first understanding your existing local admin landscape can make implementing a Privilege management solution a somewhat lengthy process.

Fortunately however, by using user Rights Discovery Mode (RDM), enterprises can now quickly understand which applications and OS features require elevated permissions across the entire desktop estate, to simplify and quicken roll out.

Introducing Rights Discovery Mode

RDM enables system administrators to monitor, analyze and concurrently report on tens of thousands of endpoints to identify what applications and tasks require administrative privileges. RDM generates reports, which can be grouped by user, computer or application to make it easy to identify trends. Applications and tasks identified by RDM can be quickly added to AM-URM configurations with just a few clicks.

Default/STANDARD rights policies enables customers to simply identify the target application that requires an elevation policy and automatically assign the application to an elevation rule to further simplify the configuration and quicken the implementation of AM or AM-URM.

Below in figure 1, is a single computer screenshot of a RDM report, grouped by application, with the administrator very quickly and easily adding an application that has been identified as requiring admin rights to a Default Rights Elevation Policy for all users. This can also be done by file name, or for extra security, by utilizing a SHA1 hash for only that specific application.

Note: Referring to the ‘List View’ (top right of the screenshot), it can be seen how many times each application was run, and from there you can also see on which machines it was run. This application details view provides you with the opportunity to see the user, machine, command line and start time for each instance of that application.

Power users often exist in enterprise environments, AppSense Application Manager provides the system administrator with the option to give selected (power) users the ability to self-elevate their rights from their default standard user account to that of a local administrator. This enables these important users to remain productive and reduces support desk calls and associated costs in relation to end user requests.

Self-elevation

Figure 1 - User Rights Discover Mode Screenshot

Figure 2- Self-elevation of user rights option
Simple user experience
The self-elevation process is extremely simple for the user and integrates seamlessly with the Windows desktop; the user simply right-clicks on the application in question and selects “Run with Administrative Rights (Audited)” from the context menu.

Granular control of self-elevation
To provide the system administrator with granular control, this feature is extremely configurable and is able to dynamically modify the configuration based on the user, device, location and time and date context.

Who can self-elevate and when
Self-elevation can be granted based on many of the existing AppSense Application Manager rules, including: user name, user group, device name and IP address. This provides a dynamic configuration that automatically changes in relation to the context of the user, device, location and time and date to ensure self-elevation only occurs when permitted and specified by the system administrator.

What a user can self-elevate
When providing the user with the option to self-elevate their rights, you may wish to restrict this ability to certain applications or actions. To achieve this AppSense Application Manager provides the system administrator with two options that can be used together to provide enhanced granularity and control:

- Whitelist - Specify the applications a user is authorized to self-elevate.
- Blacklist - Specify the applications a user is prohibited to self-elevate.

For example, the system administrator may wish to create a configuration that whitelists Notepad.exe as an authorized application and blacklists RegEdit.exe as a prohibited application.

Auditing & reporting of self-elevation
A common problem for the IT support team is clarifying “who” and “why” users require administrative rights. The reporting options within AppSense Application Manager can help with the rights discovery process.

Self-justification
When a user chooses to self-elevate their privileges to administrator and is authorized to do so, there is an option to ask the user to provide a reason as to why they are self-elevating their rights. A message alert is presented to the user with a simple text box to enter their reason.

Centralized reporting and events log
The centralized reporting and events log also captures the user name, application name, time and date of the self-elevation request and the machine it is made from. It also contains any self-justification messages the user has provided.
AppSense Application Manager and internal App stores

AppSense Application Manager enables organizations to develop their own internal, private application stores and enable selected users to introduce applications as they require them, without having to place the burden on IT and await the request to be fulfilled.

How and why
As part of the when and why a process or application is elevated, AppSense Application Manager provides the option for the system administrator to specify network drives or known good locations which can be set up so that when the user attempts to install an application from said location, AppSense Application Manager elevates the rights for the installation to enable a secure desktop but allowing known good applications to be delivered via user self-service.

This not only improves the end user experience as they are now able to self-service and fulfill their own IT service delivery requests, but it also reduces IT service costs and allows IT to focus on the delivery of the core business and corporate applications.

Granular control
Departmental app stores can be set up by taking advantage of the user or user group rules so that depending on user name or group membership they are either able or unable to install the applications in relation to which app store they are accessing.

Device based rules also ensure that the predefined applications housed in the authorized known good locations can only be installed onto permitted (e.g. corporate) devices.

Secure dialog boxes

When an application has been launched with elevated rights, it is imperative that this application cannot be used as a gateway to access the underlying desktop, operating system, other applications or files under the context of the administrator.

Secure dialog box control
The AppSense Application Manager secure dialog box control capability ensures that only the application is elevated to administrator and any dialog box such as File Open or File Save, which access the complete file structure of the computer, remains under the context of the standard user, preventing them from accessing or modifying restricted files and folders.

In the below example in figure 5, Microsoft Word is elevated to administrator, although access through a secure dialog box must not also be run under the context of the administrator. This prevents the user navigating to areas of the operating system or spawning new processes under the context of an elevated local administrator.

Thinking inside the box
The secure dialog boxes functionality is on by default and protects the system and data, stopping the user from maliciously or accidentally altering files. This includes:

- Renaming or deleting files not owned by the current user
- Dragging and dropping files to and from locked down directories
- Copying and pasting to and from locked down directories
- Changing the security permissions on a file

Much like the secure dialog boxes, it is imperative that any new process spawned from an elevated process or application does not inherit the elevated privileges, regardless of whether the spawned process is a manually launched by the user or not. If this were to happen, it would be a major security risk as it leaves the application and desktop vulnerable to misuse while the app is elevated.

Example problem: An application is raised to have administrative privilege and the user navigates to the File > Open dialog. The user browses the file system to cmd.exe and launches the application. This application is now launched as a child process of the elevated Word process.

How AppSense Application Manager solves: Without child process control, this application would now also run as a local administrator where the user could cause severe operating system damage.

AppSense Application Manager avoids this security risk by providing the option for the system administrator to prevent any child processes from adopting the elevated rights from the parent process.

Child process control

8

Figure 5 - A dialog box showing access to the file system from within an application
How it works: AppSense Application Manager and Local Security Authority (LSA) Tokens

In a Microsoft Windows computing environment when an application execution request is made, the application requests a security ticket as part of the application launch approval process. This ticket details the rights and permissions given to the application and these rights can be used to interact with the operating system or other applications.

When AppSense Application Manager is configured to manage an application, the security ticket that is requested will be dynamically modified to have permissions elevated or lowered, and therefore the application can now either run (or be blocked), with elevated or reduced privileges.

Some control panel applets, including network adapter functions are typically controlled by the explorer shell process. Elevating explorer.exe to run in the context of a local administrator is definitely not recommended as this opens the entire operating system to attack – compromising the entire purpose of least privilege. To effectively resolve this and enable the user to access the applet functionality under the context of an administrator, AppSense Application Manager provides the user with an AppSense spawned window containing the control panel applet, which can now be controlled at an access level specific to the function, without changing any of the rights associated with the explorer shell.

AppSense Privilege management integrates seamlessly with existing security principals, and does not create any new local administrator accounts, which are then referenced or used as part of applying an elevated right to a defined application.

AppSense Application Manager intercepts execution calls using detoured hooks, which are called before the real application (in question) is started. AppSense Application Manager then examines the parameters of the application request with a series of rules checks to determine if they match any of the heuristic rules and policies within the AppSense Application Manager configuration. If true, AppSense Application Manager then requests a unique access token for only the specific process from the Local Security Authority (LSA).

Note: Unlike other solutions, AppSense Application Manager ensures system security, as it does not modify the primary user token.

Now, AppSense Application Manager, with its custom authentication package impersonates the newly created unique token with administrative rights from the administrators group member Security Identifier (SID). AppSense Application Manager then holds the primary token and injects the new token, for the specific process only. This also enables AppSense to control secure dialog boxes and child process securely.

Improper control of Privilege management is serious issue in many organizations today that creates undue business risk, and significantly adds to the support cost.

Analysis by Gartner highlights that companies can save up to $1278* with effective control of privileges and reduction in admin rights.

AppSense Application Manager and AppSense Application Manager User Rights Management Edition deliver IT with more flexibility and the ability to solve access and privilege issues on a granular basis, rather than using a sledge hammer approach.

This greater control ultimately reduces desktop management costs, security risks and service requests, and improves the end user experience and employee productivity.

For further information, please visit www.appsense.com/products/desktop/desktopnow/

* - Gartner, Desktop Total Cost of Ownership, 2011 update, ID: 1470342
# Functional Matrix

Below is a functionality matrix detailing the capabilities of AppSense Application Manager as then covered in this document:

<table>
<thead>
<tr>
<th>Rights discovery</th>
<th>Contextual control of when elevation occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>User rights discovery mode</td>
<td>Define individual or groups of applications, control panel applets and system tasks</td>
</tr>
<tr>
<td>Ability to identify existing applications that make administrative calls to the operating system</td>
<td>Configurable rules based on context of user, device location and time/date</td>
</tr>
<tr>
<td>Grouping or user rights discovery results</td>
<td>Dynamic rules based on IP address of device or location of application</td>
</tr>
<tr>
<td>By application, machine or user</td>
<td>Permit elevation to only occur on predefined devices</td>
</tr>
<tr>
<td>Ability to manage applications identified in discovery reports</td>
<td></td>
</tr>
<tr>
<td>Quickly add applications to configuration to speed up implementation</td>
<td></td>
</tr>
</tbody>
</table>

### What can be elevated

- **Elevate rights for applications**: Create white and blacklists for individual or groups of applications
- **Elevate rights for control panel applets**: Create white and blacklists for individual or groups of control panel applets
- **Elevate rights for system tasks**: Create white and blacklists for individual or groups of system tasks

### Contextual control of when reduction occurs

- **Contextual control of what is reduced**: Define individual or groups of applications, control panel applets and system tasks
- **Contextual control of when reduction occurs**: Configurable rules based on context of user, device location and time/date
- **Contextual control of in what locations reduction occurs**: Dynamic rules based on IP address of device or location of application
- **Contextual control of on which devices reduction occurs**: Enforce reduction to always occur on predefined devices

### What can be reduced

- **Reduce rights for applications**: Create white and blacklists for individual or groups of applications
- **Reduce rights for control panel applets**: Create white and blacklists for individual or groups of control panel applets
- **Reduce rights for system settings**: Create white and blacklists for individual or groups of system tasks

### Internet controls and web installations

- **Enable website level application installations**: Specify trusted websites where users can install from
- **Enable application specific installations from defined websites**: Within trusted website, specify only the authorized applications users can install
- **Assign Trusted Ownership to application installation**: To enable seamless integration with AppSense Trusted Ownership application access
- **Provide pre-configured website templates**: Simplify and quicken implementation with out of the box templates

### Self-elevation

- **User self-elevation**: Enable authorized users to self-elevate without IT service request
- **User justification for self-elevation**: User is prompted to provide audited reason as to why self-elevation
- **Auditing of self-elevation**: Report on who is elevating, what they are elevating and on what device
- **Contextual control of who can self-elevate**: Based on user name, group or prelive directory
- **Contextual control of what can be self-elevated**: White or blacklist what can be self-elevated
- **Contextual control of when self-elevation can occur**: Based on time and day or other contextual rules
- **Contextual control of where self-elevation can occur**: Based on device location or where the application resides
- **Contextual control of which devices self-elevation can occur from**: Permit any nominated devices to allow self-elevation
Differences between AppSense Application Manager and AppSense Application Manager – User Rights Management Edition

<table>
<thead>
<tr>
<th>Capabilities</th>
<th>AppSense Application Manager URM Edition</th>
<th>AppSense Application Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>User rights discovery mode</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Contextual control of rights management and invocation</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Granular metadata-based configuration option</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Web and application level installation controls</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>User self-service and self-elevation options</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Control of dialogue boxes and child-processes</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Digital signature validation controls</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Auditing and reporting capabilities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Application whitelisting and black listing for user rights</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Application whitelisting and black listing for app access</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Device license enforcement</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Dynamic application access entitlement</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Block untrusted code execution (Trusted Ownership)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Context aware application termination</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Archiving of unauthorized or self-approved applications</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Granular URL filtering and re-direction options</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Network access control</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Application time limits</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Number of application instance controls</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Script, Batch File and Registry controls</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Passive reporting for user introduced executable code</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>End point analysis and application usage scans</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Pre-built AppSense Configuration Templates</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
About AppSense

AppSense, the people-centric computing company, is the leading provider of user virtualization technology that transforms organizations into productive mobile workforces securely governed by IT. AppSense enables companies to embrace consumerization in the enterprise by independently managing all aspects of the user experience across mobile devices and desktops. Our user virtualization technology reduces IT complexity and improves the deployment, management and migration of multi-platform desktop and mobile environments. The company is headquartered in New York, NY with offices around the world. For more information visit www.appsense.com